

IN THE CLAIMS

The following listing of claims replaces all previous claim listings and versions:

1-32 Cancelled

33. (Currently amended) A software ensemble stored on a computer readable medium and executable by a computer, the software ensemble comprising:
a plurality of software units, each software unit of said plurality of software units including a method and data;
an executive software unit including links between said plurality of software units

~~The software ensemble according to claim 27, wherein a software unit of the software ensemble is described by a model M , given by:~~

$M = (inGates, \{inSign_g\}, \{a_g\}, Q, q_0, outGates, \{outSign_{gt}\}, \{outFunction_{gt}\})$,
where $inGates$ is the set of software unit input gates, $outGates$ is the set of software unit output gates, a_g is an action for every g in $inGates$, Q is the set of software unit states, q_0 is the software unit's initial state, $inSign_g$ is the gate gt input-output signature for each gt in $inGates$, $outSign_{gt}$ is the gate gt output-input signature for each gt in $outGates$, and $outfunction_{gt}$ is the gate gt output function for every gt in $outGates$, where the signature is a 2-tuple containing the range set of incoming and outgoing parameters, and the output function of gate gt maps a set of incoming values into a value belonging to the input signature of the output gate gt .

34. (Currently amended) A software ensemble stored on a computer readable medium and executable by a computer, the software ensemble comprising:
a plurality of software units, each software unit of said plurality of software units including a method and data;
an executive software unit including links between said plurality of software units

~~The software ensemble according to claim 27, wherein the software ensemble, E , is defined by:~~

$$E = (inGates, \{inSign_g\}, \varepsilon, M_\varepsilon, outGates, \{outSign_{gt}\}, \{outFunction_{gt}\}),$$

where ε is the ensemble executive that keeps a structure of the ensemble, and M_ε is the model of the ensemble executive, $inGates$ is the set of the ensemble software unit input gates, $inSign_g$ is the gate g input-output signature for each g in $inGates$, $outGates$ is the set of ensemble software unit output gates, $outSign_{gt}$ is the gate gt output-input signature for each gt in $outGates$, and $outFunction_{gt}$ is the gate gt output function for every gt in $outGates$, where a signature is a 2-tuple containing the range set of incoming and outgoing parameters, and the output function of gate gt maps a set of incoming values into a value belonging to the input signature of output gate gt .

35. (Currently amended) The software ensemble according to claim 34, where the model of the ensemble executive is defined as a model of a reusable software unit augmented with a structure function $\sigma: Q \rightarrow \Sigma^*$, where Q is the state set of the ensemble executive, and each structure Σ in Σ^* is equal to $(C, \{M_c\}, L, \Xi)$, and

where C is the set of reusable software units that belong to the ensemble, M_c is the definition of each reusable software unit c , belonging to set C , L is a set of channels, and Ξ is the order function indicating a sequence in which actions are invoked.

36. (Previously presented) The ensemble executive according to claim 35, wherein a channel is a 3-tuple defined by:

$$((i, g_i), (j, g_j), (dF, rF)),$$

where i is the name of a sender software unit, g_i is a gate of i , j is a receiver software unit, g_j is a gate of j , dF is the channel direct filter and rF is the channel reverse filter, where a filter defines a transformation that is applied to the values communicated through a channel.

37-50 Cancelled

51. (Currently Amended) The software ensemble according to claim 35 34, wherein anthe input gate and the channels determine the ~~one or more of the~~ methods invoked at the one or more of the plurality of the software units when a message is received by the input gate of the software ensemble.

52. (Previously presented) The software ensemble of claim 35, wherein the output gate of a software unit and the channels determine the ~~one or more of the~~ messages to be sent through the output gates of the software ensemble.

53. Cancelled

54. (New) The software ensemble according to claim 36, wherein when a value x is sent from gate g_i to gate g_j by a channel $((i, g_i), (j, g_j), (dF, rF))$, the software unit j performs $j \ a_{g_i} \ dF(x)$ and returns the value $rF(j \ a_{g_i} \ dF(x))$, where a_{g_i} is an action associated with the gate g_i of the receiver software unit.

55. (New) The software ensemble according to claim 54, wherein when the receiver software unit is an ensemble, the value $dF(x)$ is passed to the software units connected to the input port g_j according to channel information defined for that ensemble.

56. (New) The software ensemble according to claim 34, wherein structural inheritance is utilized to build new software ensembles from existing software ensembles.